High Maturity Is Not A Procrustean Bed

Barry Boehm, Stevens Institute of Technology
Richard Turner, Stevens Institute of Technology
Jo Ann Lane, University of Southern California
Supannika Koolmanojwong, University of Southern California

Abstract. In Greek mythology, Procrustes was a rogue smith and bandit who invited travellers to rest in his "perfectly sized bed." When they accepted, he forcibly bound them to it, then stretched them or cut off various body parts until they "perfectly" fit the bed. Too many organizations have a single model of high maturity to which they try to fit all their projects. Development and acquisition organizations are finding that competitive success requires systems that are a mix of high security assurance components, opaque and dynamic COTS products and cloud services, and highly useful but kaleidoscopic apps and widgets. Approaching such systems with a one-size-fits-all corporate process and maturity model often results in a procrustean fit.

As a process model generator, the Incremental Commitment Spiral Model has a set of criteria for determining which process or processes best fit a particular system of interest. This article summarizes the criteria and illustrates how they have been successfully applied in various situations [1].

Introduction

Too often, high maturity is seen as a proven, standard process that is tailored down or up or in other ways twisted and tortured to adapt to projects that simply don't fit the process. This flies in the face of the definition of a high maturity organization as agile, flexible, and continuously improving. Rapid change, requirements uncertainty, and short capability delivery cycles are increasing the need for such agility, and the traditional process and lifecycle models are not meeting the challenge.

Table 1 describes some examples of Procrustean situations that result from inflexible or overly constrained "high maturity" or otherwise "disciplined" approaches. It elaborates the situation into the likely undesired project result, an example, and a remedy or means of avoiding the situation using the ICSM's four primary principles: Stakeholder value-based guidance; Incremental commitment and accountability; Concurrent multi-discipline engineering; and Evidence and risk-based decisions.

A Different Approach

The Incremental Commitment Spiral Model (ICSM),[1,2] shown in Figure 1, is the result of our efforts to better integrate the hardware, software, and human factors aspects of systems, to provide value to the users as quickly as possible, and to handle the increasingly rapid pace of change. While its pedigree lies in the spiral concept first broadly published in 1988,[3] this new version draws on over 20 years of experience helping people deal with the fact that the original version was too easy to misinterpret.

Fundamental Principles

In hindsight, most of the problems in using the 1988 spiral model came from users constructing processes that had nothing to do with the underlying concepts. The ICSM's four underlying principles, based on observed failure modes over years of experience, are:

Stakeholder value-based guidance. Failing to include and address the value propositions of its success-critical stakeholders can result in their minimal commitment to the project; they may underperform, decline to use, or block the use of the results.

Incremental commitment and accountability. If success-critical stakeholders are not accountable for their commitments (or lack thereof), and the associated consequences (good or bad), they may not provide necessary commitments or decisions in a timely manner and are likely to be drawn away to other pursuits when they are most needed.

Concurrent multi-discipline engineering. Sequential definition and development of a) requirements and solutions; b) hardware, software, and human factors; or c) product and processes likely slows the project and leads to early, hard-to-undo commitments that limit options for project success.

Evidence and risk-based decisions. If key decisions are made based on assertions, vendor literature, or meeting an arbitrary schedule without access to evidence of feasibility, the project is building up risks.

The annual series of "Top-5 Quality Software Projects" software-intensive systems projects published in CrossTalk[4] are examples of successful projects that applied the ICSM principles. These were chosen annually between 2002 and 2005 by panels of leading experts as role models of best practices and successful outcomes. Of the 20 Top-5 projects, 16 explicitly used concurrent engineering; 14 explicitly used risk-driven development; and 15 explicitly used incrementally committed, iterative system evolution. Additional projects gave indications of their partial use. Unfortunately, the project summaries did not include discussion of stakeholder involvement.

The ICSM is not a single one-size-fits-all process. It is actually a process generator, which steers your process in different directions, depending on your particular circumstances. Unlike in the traditional sequential approaches, each spiral concurrently addresses all of the activities of product development to include:

- Requirements (objectives and constraints)
- Solutions (alternatives)
- Products and processes
- Hardware
- Software
- Human factors aspects
- Business case analysis of alternative product configurations
- Product line investments

In this way, ICSM helps adapt your lifecycle strategies and processes to your sources of change. It also supports more rapid system development and evolution through concurrent engineering, enabling you to develop and evolve systems more rapidly and to avoid obsolescence. It is, in many ways, the antithesis of Procrustes bed – one that adjusts to the person, not the other way around.

Report Documentation Page		*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **JUL 2014**	2. REPORT TYPE	3. DATES COVERED **00-00-2014 to 00-00-2014**
4. TITLE AND SUBTITLE **High Maturity Is Not A Procrustean Bed**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Stevens Institute of Technology,Castle Point On Hudson,Hoboken,NJ,07030**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		

14. ABSTRACT

In Greek mythology, Procrustes was a rogue smith and bandit who invited travellers to rest in his ?perfectly sized bed.? When they accepted, he forcibly bound them to it, then stretched them or cut off various body parts until they ?perfectly? fit the bed. Too many organizations have a single model of high maturity to which they try to fit all their projects. Development and acquisition organizations are finding that competitive success requires systems that are a mix of high security assurance components opaque and dynamic COTS products and cloud services, and highly useful but kaleidoscopic apps and widgets. Approaching such systems with a one-size-fitsall corporate process and maturity model often results in a procrustean fit. As a process model generator, the Incremental Commitment Spiral Model has a set of criteria for determining which process or processes best fit a particular system of interest. This article summarizes the criteria and illustrates how they have been successfully applied in various situations [1].

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**7**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Table 1. Examples of Procrustean Process Consequences

Issue	Result	Example
Defined process mismatch	Large systems get their integration lopped off in trying to keep to 4-week increments	In their paper "Recognizing and Responding to 'Bad Smells' in Extreme Programming," Amr Elssamadisy and Gregory Schalliol of ThoughtWorks describe a case where after 3 years of success with applying XP to a lease management system, the length of time to add a new feature became longer than an iteration due primarily to increasingly complex integration and technical debt issues.
Poor contracting	Development lopped off by a fixed-price, fixed SOW contract	TRW spent money and schedule designing a system to a 1-second response time requirement only to find that this was not affordable. Luckily, this was discovered early and only cost 13 months of schedule.
Policy influences (on standards development)	Stretched requirements result in wasteful expenditures on non-value adding work that stretch schedules and budgets	The definition of MIL-STD-498 as a replacement for 2167 and 7935. Wanting to avoid imposing 23 DIDs that on simple could be tailored down but in principle rarely were in practice, two other versions (one with 6 DIDs and one with 1 DID) were developed. The policy police decided that DoD couldn't have more than one set of documents covering the same content, leaving only the 23 DID version.
Policy influences (Expert-developed standards)	Lack of understanding, "short" sighted policy definition and "long" impacts leading to disastrous process implementations	The framers of 2167 and 2167A didn't see the waterfall diagrams as a problem, because "anyone with common sense would know better than to commit to requirements without establishing their feasibility." But less-expert project managers would see" following the standard" as the safest thing for their careers, and end up getting into trouble.
Policy Influences (Piling On Constraints)	Rework and technical debt overload stretch schedules and budgets	Changing the rules mid-stream with inflexible processes is disastrous. An organization started with the 2167 mandate to have the requirements determine the delivered capabilities. mid-way through the project, a SecDef memo to "use COTS products wherever possible," meant COTS capabilities would determine the requirements. Later in the development, a mandate to use the Ada programming language resulted in significant effort because many of the selected COTS products had weak or no Ada bindings.
Top-Executive Mandates	Unintentionally imposed constraints that cut off technical solution options	Dated executive experience often constrains their decisions. A 2006 Mark Maier SysE Journal paper identified hardware architecture constraints imposed by hardware-oriented top executives in terms of functional hierarchies and simple interfaces that cut off software options such as layered-service architectures and more complex but necessary interface protocol compatibility standards options.
Voice of the Customer.	Every customer need becomes a project requirement, stretching the project well outside budget and schedule constraints	The Bank of America Master Net project used a broad, unmediated Voice of the Customer approach that ended up in a disaster when the major stakeholders' agreed-to desires resulted in significant success model clashes and overruns.
Test-Driven Acceptance.	Under-constrained acceptability, leaving extremities to be lopped off later	The 3000-test Ada compiler validation suite led compiler vendors to patch their compiler software to pass the tests, creating a product that was often less robust than their beta-test versions.
Search-Driven Acceptance.	Projects deploying inappropriate practices, methods or approaches	Search engine results on the use of formal methods found mostly success stories, but on small projects, leading some projects to adopt the methods only discover scalability shortfalls.
Auditor-Driven Acceptance.	Varying auditor interpretations over constrain or under constrain projects leading to stretching or chopping later	Software CMM or CMMI auditor-based maturity levels requirements had little impact on acquisition programs.
Value-Neutral Acceptance.	Inappropriate activity and gaming on the part of developers driven by simplistic or incomplete metrics	Some projects use delivered defect density as the basis for acceptance, leading project personnel to fix the easy defects. The project then finds the hard defects are unacceptable, and must be stretched well beyond its budget to become acceptable.
Acquisition-Oriented Acceptance.	Product too expensive to operate and maintain	Tight budgets and schedules lop off options to design and develop the project to facilitate maintenance, operations, and support.

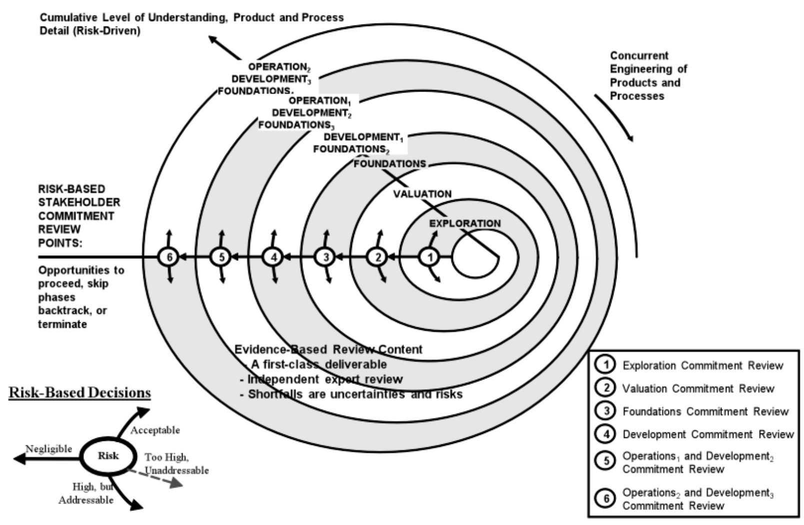

Cumulative Level of Understanding, Product and Process Detail (Risk-Driven)
OPERATION₂ DEVELOPMENT₃ FOUNDATIONS₄
OPERATION₁ DEVELOPMENT₂ FOUNDATIONS₃
DEVELOPMENT₁ FOUNDATIONS₂
FOUNDATIONS
VALUATION
EXPLORATION
Concurrent Engineering of Products and Processes
RISK-BASED STAKEHOLDER COMMITMENT REVIEW POINTS:
Opportunities to proceed, skip phases backtrack, or terminate
Evidence-Based Review Content
- A first-class deliverable
- Independent expert review
- Shortfalls are uncertainties and risks
Risk-Based Decisions
Acceptable
Negligible
Risk
Too High, Unaddressable
High, but Addressable
1 Exploration Commitment Review
2 Valuation Commitment Review
3 Foundations Commitment Review
4 Development Commitment Review
5 Operations₁ and Development₂ Commitment Review
6 Operations₂ and Development₃ Commitment Review

Figure 1. The Incremental Commitment Model: Spiral View

ICSM Lifecycle

The Phased View (Figure 2) shows how the overall lifecycle process divides naturally into two major stages. Stage I, Incremental Definition, covers the up-front growth in system understanding, definition, feasibility assurance, and stakeholder commitment. If the Phase I activities do not result in deciding to radically change the effort by adjusting scope or priorities, or discontinuing the development completely, they lead to a larger Stage II commitment to implement a feasible set of specifications and plans for Incremental Development and Operations.

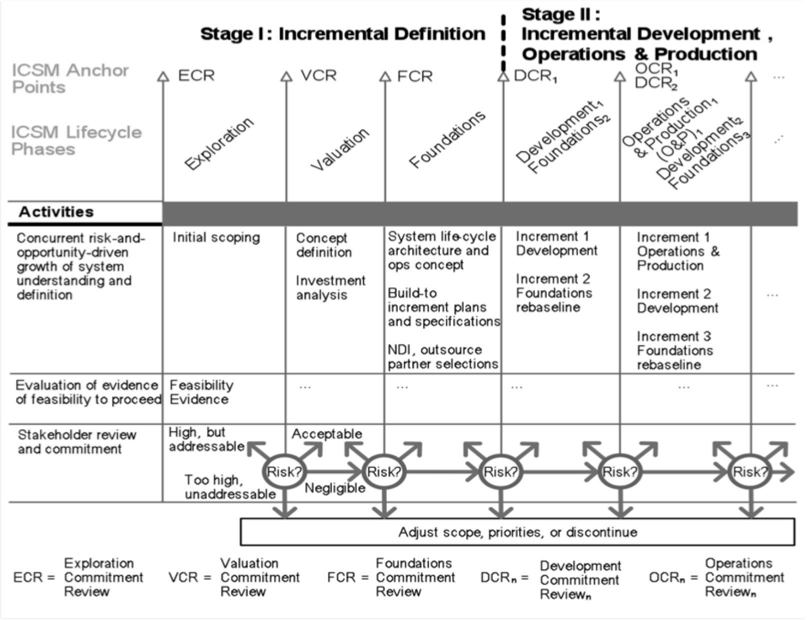

Stage I: Incremental Definition
Stage II: Incremental Development, Operations & Production
ICSM Anchor Points
ECR
VCR
FCR
DCR₁
OCR₁ DCR₂
...
ICSM Lifecycle Phases
Exploration
Valuation
Foundations
Development₁ Foundations₂
Operations & Production₁ (O&P)₁ Development₂ Foundations₃
Activities
Concurrent risk-and-opportunity-driven growth of system understanding and definition
Initial scoping
Concept definition
Investment analysis
System life-cycle architecture and ops concept
Build-to increment plans and specifications
NDI, outsource partner selections
Increment 1 Development
Increment 2 Foundations rebaseline
Increment 1 Operations & Production
Increment 2 Development
Increment 3 Foundations rebaseline
Evaluation of evidence of feasibility to proceed
Feasibility Evidence
Stakeholder review and commitment
High, but addressable
Acceptable
Too high, unaddressable
Negligible
Risk?
Adjust scope, priorities, or discontinue
ECR = Exploration Commitment Review
VCR = Valuation Commitment Review
FCR = Foundations Commitment Review
DCRₙ = Development Commitment Reviewₙ
OCRₙ = Operations Commitment Reviewₙ

Figure 2. The ICSM Staged View

The duration of Stage I can be anywhere from one week to five years, depending on factors like the number, capability, and compatibility of the proposed system's components and stakeholders. A small, experienced, developer-customer team, using agile software methods and operating on a mature infrastructure, can form and begin incremental development of a well-defined software project in less than a week. A more complex project requires significant effort and could take up to five years or more. An example might be an ultra-large, unprecedented, multi-mission, multi-owner, system-of-systems needing to integrate with numerous independently evolving legacy or external systems. We have provided ICSM elements to the definition and development of such systems.[5]

Stage II is planned around the length of the increments to be used in the system's development and evolution. This is a key decision made during the Development Commitment Review. A small agile project can use two-to-four week increments. A much larger project could need increments of up to two years to develop and integrate an increment of operational capability. However, the ICSM capability delivery cadence is not necessarily linked to the internal development cadence, and there may be several internal integration cycles within a longer release increment. Some large, inseparable, hardware components would take even longer to develop their initial increments, and would be scheduled to synchronize their capability deliveries with concurrently evolving infrastructure or software increments.

Stage I activities have assured a common vision, committed stakeholders, and an architecture capable of accommodating foreseeable changes such as user interfaces, external system interoperability requirements, or transaction formats. These enable the features in each Stage II increment to be prioritized and the increment timeboxed.

Flexible, Multiple and Evolving Processes

The ICSM essentially uses evidence and risks to generate appropriate processes throughout the lifecycle. Figure 3 illustrates four example paths through the ICSM to visualize how different risks create different processes.

Example A is a simple business application based on an already-available Enterprise Resource Planning (ERP) package. There is no need for a Valuation or Architecting activity if the ERP package has already been purchased and its architecture has already proved cost-effective in supporting more complex applications. Thus, the project can go directly into Stage II, using an agile method such as a combination of Scrum and Extreme Programming. There is no need for "Big Design Up Front" activities or artifacts because an appropriate architecture is already present in the ERP package. Nor is there a need for heavyweight waterfall or V-model specifications and document reviews. The critical risk identified at the end of Exploration could be the user acceptance and business process reengineering required for deployment. In this case, that risk would be considered negligible if the system's human interface risks have been sufficiently mitigated via ERP package-based prototyping.

Example B involves a risky but innovative system such as adding a retina scanner to the next model of a cellphone product. There are a number of uncertainties and risks/opportunities to resolve, such as scanner hardware integration and safety of the user. But the new capability is needed quickly and there is a fallback (deferring its introduction to the following model), so proceeding to address the risks and develop the system is acceptable.

Example C is a system that is defined as safety critical. The stakeholders responsible for the safety of the proposed system find at the Foundations Commitment Review that the proposers have provided inadequate safety evidence. It is better to have the proposers develop such evidence through architecture-based safety cases, fault tree analyses, and failure modes and effects analyses before proceeding into the Foundations phase. The arrow back into the Valuation phase indicates this.

In Example D, the developers are simply too late to play. It is discovered before entering the Development phase that a superior product has already entered the marketplace, leaving the current product with an infeasible business case. Here, unless adjusting the project's scope can make a viable business case, it is best to discontinue it. It is worth pointing out that it is not necessary to proceed to the next major milestone before terminating a clearly non-viable project; however, stakeholder concurrence in termination is essential.

ICSM Risk-Driven Common Cases

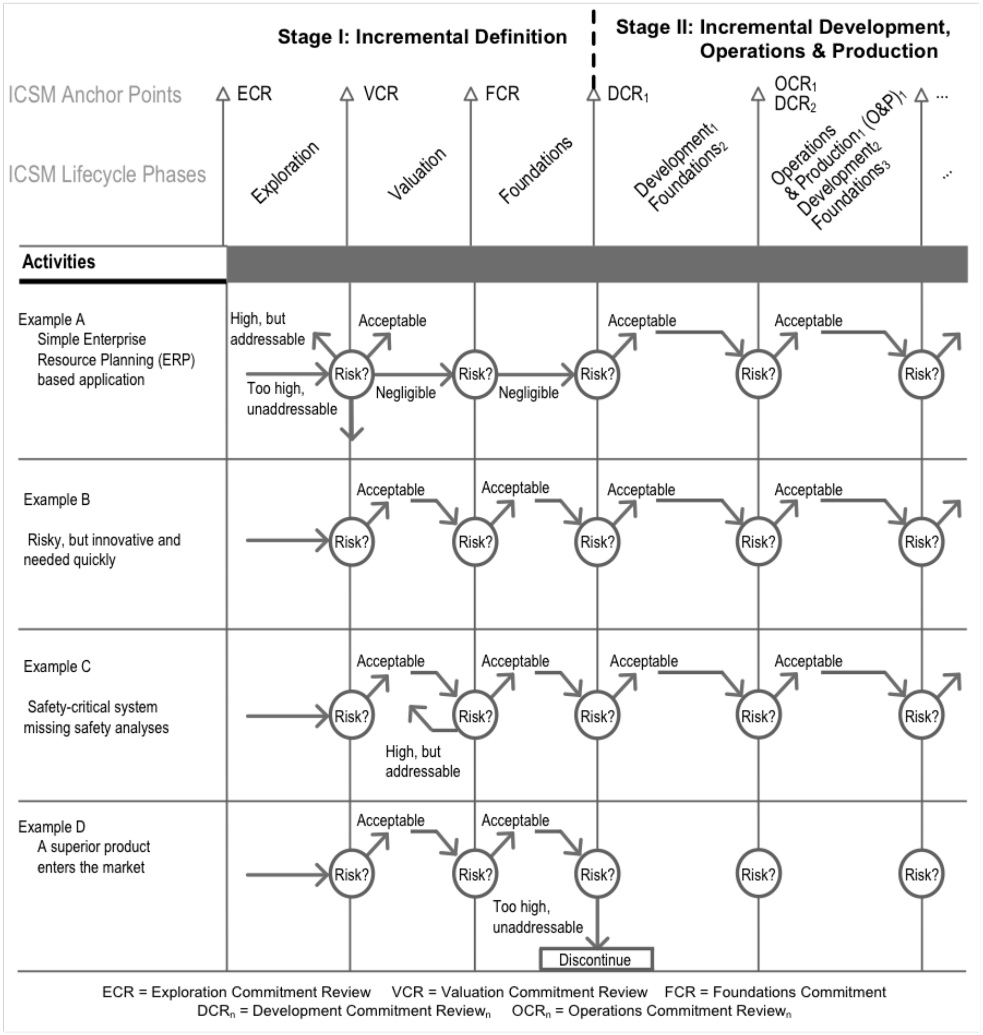
Many projects can reuse experience from previous projects. However, every project has the possibility of unique aspects that could impact the selection of processes and the path through the ICSM. To enable early estimation, supply examples that help users with initial planning, and support categorization and capture of lessons learned, we have identified a set of seven risk patterns that represent the most often seen paths through the ICSM. We have named these patterns Common Cases:

- Software application or system
- Software-intensive device
- Hardware platform
- Family of systems or product line
- System of systems (SoS) or enterprise-wide system
- Brownfield modernization

Table 2 briefly describes when to use each common case and some examples of each.

ICSM and Large, Complex Systems

Obviously, larger, more complex systems will require a great deal more activity in Stage I. In Stage II, however, the ICSM allows a great deal of flexibility in providing a way of integrating and accommodating the wide variety of development activities that can appear across the various hardware, software, and human development activities. For that reason, the Implementation Phase is based on a three-tiered, timeboxed process that allows for reflection, anticipation, and adjustment to the changing environment, shown in Figure 4. This concept works best in software, but can apply to hardware in many cases. Figure 5 shows how this three-tiered model scales to multiple component or subsystem development.



Stage I: Incremental Definition
Stage II: Incremental Development, Operations & Production
ICSM Anchor Points
ECR
VCR
FCR
DCR1
OCR1 DCR2
...
ICSM Lifecycle Phases
Exploration
Valuation
Foundations
Development1 Foundations2
Operations & Production1 (O&P)1 Development2 Foundations3
...
Activities
Example A Simple Enterprise Resource Planning (ERP) based application
High, but addressable
Too high, unaddressable
Acceptable
Negligible
Risk?
Example B Risky, but innovative and needed quickly
Example C Safety-critical system missing safety analyses
Example D A superior product enters the market
Discontinue
ECR = Exploration Commitment Review VCR = Valuation Commitment Review FCR = Foundations Commitment
DCRn = Development Commitment Reviewn OCRn = Operations Commitment Reviewn

Figure 3. Different Risk Patterns Yield Different Processes

System (or subsystem)	Common Case	Examples
SW application/system executing on one or more commercial HW platforms, as a standalone system or a constituent of one or more SoSs.	SW application or system	Cellphone app, business application or system, military command and control software system, inventory management systems, computer operating system, database management system
A special purpose object, machine, or piece of equipment that has significant features provided by software.	SW-intensive device	Computer peripherals, weapons, entertainment devices, health care devices (including small surgical), GPS receivers, manufacturing tools
Vehicle (land, sea, air, or space)	HW platform	Small unmanned vehicle, automobile, tank, ship, airplane, space shuttle, space station, Mars rover
Computer	HW platform	Mainframe, server, laptop, tablet, cellphone
Part of a set of systems that are either similar to each other or interoperate with each other	Family of systems or product line	Car models that share many core components; interoperating back-office systems such as billing, accounting, and sales force support, that share a common repository with standard data definitions and formats, and are provided by a single vendor
A new capability that will be performed by more than one interoperating system	SoS or enterprise-wide system	Multiple interoperating systems owned and managed by different organizations; for example, navigation systems that include airborne and land systems using GPS
Refactoring or re-implementation of an older legacy system or set of systems	Brownfield modernization	Incremental replacement of old, fragile business systems with COTS products or technology refresh/upgrade of existing systems

Table 2. ICSM Common Cases

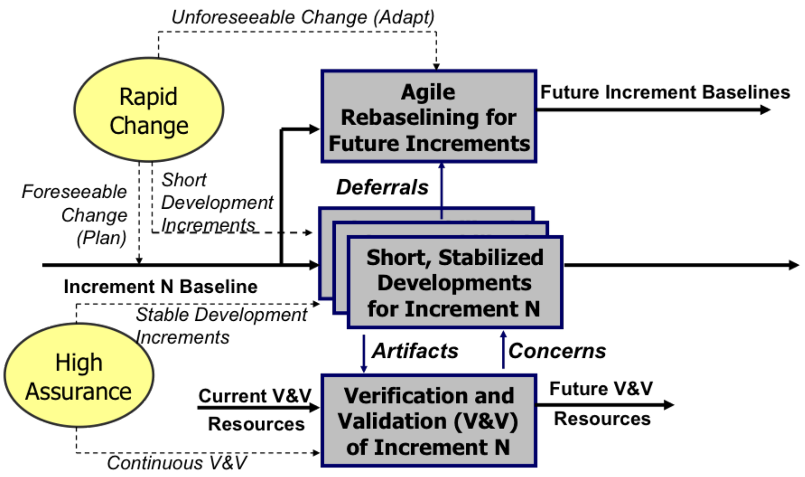

Unforeseeable Change (Adapt)
Rapid Change
Agile Rebaselining for Future Increments
Future Increment Baselines
Foreseeable Change (Plan)
Short Development Increments
Deferrals
Short, Stabilized Developments for Increment N
Increment N Baseline
Stable Development Increments
High Assurance
Artifacts
Concerns
Current V&V Resources
Verification and Validation (V&V) of Increment N
Future V&V Resources
Continuous V&V

Figure 4. Three-tier timeboxed approach (Evolution View)

ICSM and Process Improvement

ICSM is designed to provide flexibility. It also expects you to evaluate and apply the process assets you already have in new ways, and provides essential guidance on hw that can happen. ICSM also seeks to actively create and use lessons learned both within and between projects to decrease the learning cycle and accelerate improvement. The key intrinsic process improvement aspects in ICSM are evidence, risk-based process, the incremental approach, and anticipation/reflection.

In the ICSM, evidence is continuously created as a first class deliverable and used for process generation, decision-making, and stakeholder commitment. This evidence captures a wide variety of knowledge in a way that can be empirically analyzed to support retrospection at almost every point in the lifecycle. It can also be used to improve estimation, evaluate experimental processes and methods, and transfer knowledge across projects and systems.

As with many process models, risks are captured and tracked. However, in the ICSM they also directly impact the process generation activities and are integrated into all decision-making. Many risks are common across a domain, and so mitigation efforts based on ICSM process decisions are documented and can be easily captured to support decision-making and process generation across projects.

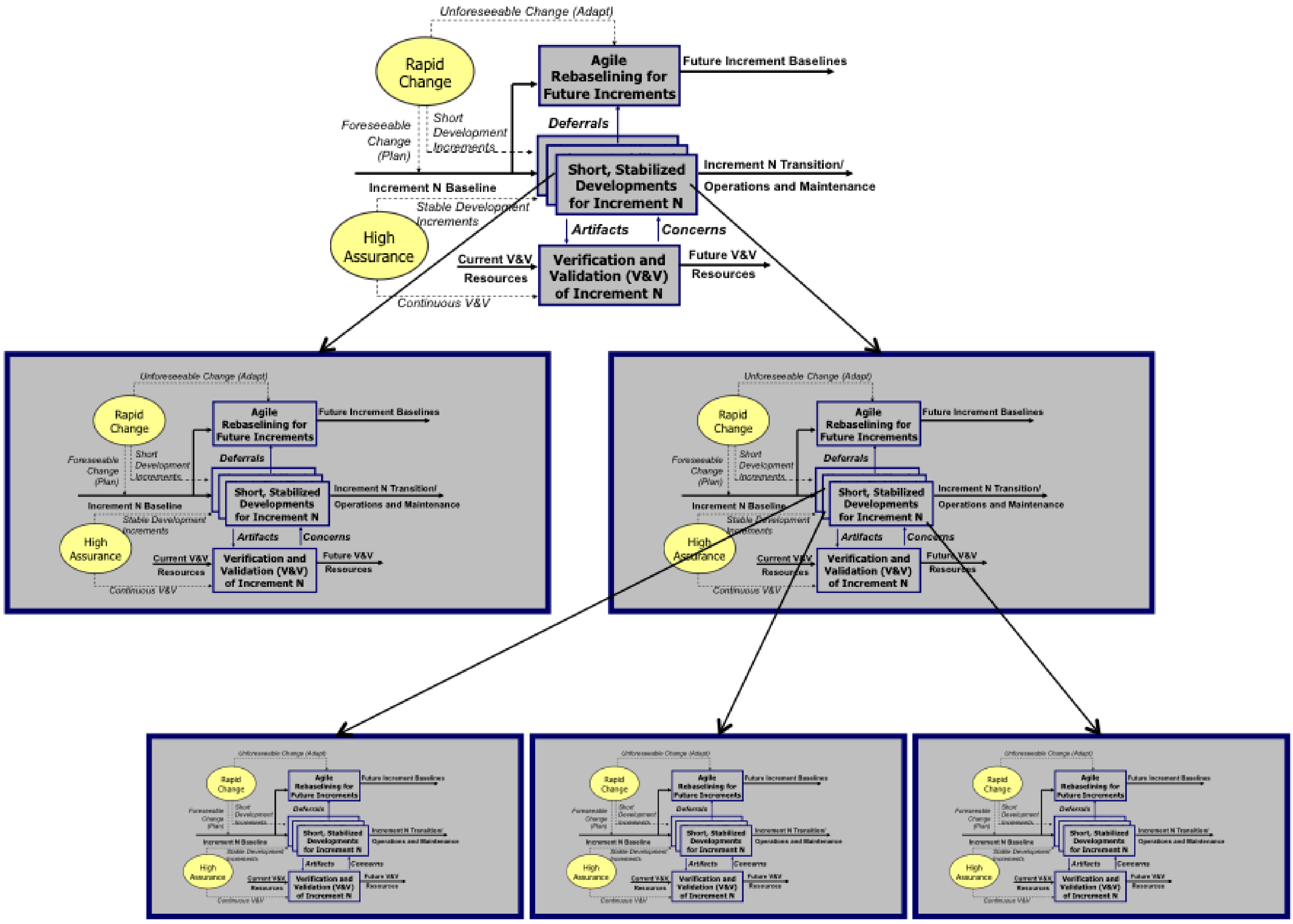

Unforeseeable Change (Adapt)
Rapid Change
Agile Rebaselining for Future Increments
Future Increment Baselines
Foreseeable Change (Plan)
Short Development Increments
Deferrals
Short, Stabilized Developments for Increment N
Increment N Transition/ Operations and Maintenance
Increment N Baseline
Stable Development Increments
High Assurance
Artifacts
Concerns
Current V&V Resources
Verification and Validation (V&V) of Increment N
Future V&V Resources
Continuous V&V

Figure 5. A Large-system development phase

Issue	ICSM Mitigation
Defined process mismatch	Track evidence of time needed to both develop and integrate new increments, and adjust increment sizes and/or schedules as necessary
Poor contracting	Develop evidence of the need for 1-second response time and the cost of achieving it before committing to it.
Policy influence (on standards development)	Develop and sustain multiple sources of guidance for deliverables on different classes of systems, such as with the recent draft update of DoDI 5000.02
Policy influence (Expert-driven standards)	Provide criteria for initial choice of project process, and risk-based decision guidance on process adaptation to change
Policy Influence (Piling On Constraints)	Add new guidance directives only based on evidence of their compatibility with existing directives
Top-executive Mandates	Involve development and support stakeholders in key process and product guidance. Concurrently engineer the system's hardware, software, and human elements
Voice of the Customer.	Involve all success-critical stakeholders in key project and product guidance decisions
Test-driven Acceptance	Evolve test criteria based on user alpha, beta-test experience
Search-driven Acceptance	Ensure that evidence is accumulated from fully representative stakeholder communities
Auditor-driven Acceptance	Involve stakeholders in choice of process and product guidance.
Value-neutral Acceptance	Use stakeholder value propositions to prioritize requirements, proposed changes, test cases, defect fixes
Acquisition-oriented Acceptance	Involve post-deployment stakeholders in determination and prioritization of requirements

Table 3.ICSM mitigations to procrustean issue

The incremental nature of the ICSM shortens the learning cycle. Agile and lean development methods with short cycle times, value-based scheduling, and continuous integration can be employed wherever appropriate. Coupled with the ICSM emphasis on evidence and risk, these can accelerate learning, reduce rework, and manage technical debt in such a way as to provide continuous process improvement throughout the Stage II activities.

Finally, process improvement requires balanced reflection and anticipation. Wayne Gretzky, who is generally acknowledged as the greatest hockey player of all time, ascribes a good deal of his success to the ability to anticipate where the hockey puck was going, and to skate to where he could capitalize on that knowledge. Anticipating where technologies, competitors, organizations, and the marketplace are going is increasingly critical to successful systems and software engineering. In contrast, organizations that spend their time asking, "How could we have done our last project better?" are actually skating to where the puck has been. Clearly, such "reflection in action" is good,[6] but in a world of rapid change, reflection in action needs to be balanced with anticipation. The Incremental Commitment Spiral Model integrates reflection, anticipation, and agility to take advantage of evolving knowledge through a risk-based, principle-driven approach to system development. We are still firm believers that there are no panaceas, silver bullets, or one-size-fits-all solutions. We are confident, though, that the ICSM offers a coherent and useful way to approach systems development in a world that has not only changed, but will also continue to change throughout every system's life cycle.

Conclusions

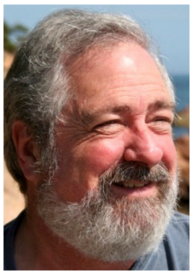
Procrustes caused a lot of damage before Theseus turned the tables (or the bed) on him. We believe that there are a lot of ways to fight procrustean tendencies through rethinking the processes we advocate, and pushing back on those who are applying inappropriate or damaging processes to our projects. One of these ways is using the process generation framework provided by the ICSM. Table 3 shows how the ICSM can mitigate our earlier list of procrustean issues.

ISCM supports adapting and applying multiple processes (or process assets) as needed throughout a project, regardless of size, duration, or complexity. It provides a flexible, extensible lifecycle that can be adopted across a wide variety of project environments. Most importantly, it establishes all of the underlying principles of high maturity organizations— stakeholder value, incrementality, concurrency, agility, flexibility, empiricism, improvement and predictability—without restricting the specific processes deployed. ICSM enables the opposite of a procrustean process: one that adapts to your needs rather than forcing you to meet its own.

ABOUT THE AUTHORS

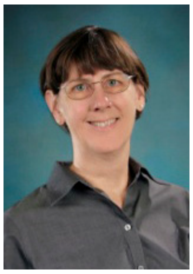
Dr. Barry Boehm is a USC Distinguished Professor and Chief Scientist of the DoD-Stevens-USC Systems Engineering Research Center,. He was director of DARPA-ISTO 1989-92, at TRW 1973-89, at Rand Corporation 1959-73, and at General Dynamics 1955-59. He is a Fellow of the primary professional societies in computing (ACM), aerospace (AIAA), electronics (IEEE), and systems engineering (INCOSE), and a member of the U.S. National Academy of Engineering.

E-mail: barryboehm@gmail.com



Dr. Richard Turner is a Distinguished Service Professor at the Stevens Institute of Technology. Active in the agile, lean and kanban communities, he helped author the Software Extension to the PMI Guide to the PMBOK. He is a Golden Core member of the IEEE Computer Society, a fellow of the Lean Systems Society and co-author of four books: The Incremental Commitment Spiral Model, Balancing Agility and Discipline, CMMI Survival Guide, and CMMI Distilled.

Phone: 202-390-3772
E-mail: rturner@stevens.edu



Jo Ann Lane is currently the systems engineering Co-Director of the University of Southern California Center for Systems and Software Engineering, a member of the Systems Engineering Research Center Research Council representing the system of systems research area, and emeritus professor of computer science at San Diego State University. Her current areas of research include system of systems engineering, system affordability, expediting systems engineering, and balancing agile techniques with technical debt.

Phone: 858-945-0099
E-mail: jolane@usc.edu

Dr. Supannika Koolmanojwong is a lecturer and a researcher at the University of Southern California Center for Systems and Software Engineering. Her primary research areas are Software Process Improvement, Software Process Quality Assurance, Software Metrics and Measurement, Agile and Lean Software Development and Expediting Systems Engineering. She is a certified scrum master and a certified Product Owner. Prior to this, she was a software engineer and a RUP/OpenUp Content Developer at IBM Software Group.

E-mail: koolmano@usc.edu

NOTES

1. Boehm, B. and J. Lane, "Using the Incremental Commitment Model to Integrate System Acquisition, Systems Engineering, and Software Engineering," CrossTalk, October, 2007
2. Boehm, B., J. Lane, S. Koolmanojwong, and R. Turner, The Incremental Commitment Spiral Model: Principles and Practices for Successful Systems and Software, Addison Wesley Pearson, New York, 2014.
3. Boehm, B. "A Spiral Model for Software Development and Enhancement." Computer. May 1988;61–72.
4. CrossTalk. "Top Five Quality Software Projects." January 2002, July 2003, July 2004, September 2005. www.stsc.hill.af.mil/crosstalk.
5. Stephen Blanchette Jr., Steven Crosson, Barry Boehm, "Evaluating the Software Design of a Complex System of Systems," CMU/SEI Tech Report CMU/SEI-2009-TR-023, January 2010
6. D. Schon, The Reflective Practitioner. Basic Books, 1983.

REFERENCES

1. Much of the material in this article is drawn from a new book: Boehm, B., J. Lane, S. Koolmanojwong, and R. Turner, The Incremental Commitment Spiral Model: Principles and Practices for Successful Systems and Software, Addison Wesley Pearson, New York, 2014. The initial work that provided the basis for the book was funded in part by the US Department of Defense, through the Systems Engineering Research Center, a University Affiliated Research Center at Stevens Institute of Technology.

U.S. DEPARTMENT OF HOMELAND SECURITY

Homeland Security

The Department of Homeland Security, Office of Cybersecurity and Communications (CS&C) is responsible for enhancing the security, resiliency, and reliability of the Nation's cyber and communications infrastructure and actively engages the public and private sectors as well as international partners to prepare for, prevent, and respond to catastrophic incidents that could degrade or overwhelm these strategic assets. CS&C seeks dynamic individuals to fill critical positions in:

- Cyber Incident Response
- Cyber Risk and Strategic Analysis
- Networks and Systems Engineering
- Computer & Electronic Engineering
- Digital Forensics
- Telecommunications Assurance
- Program Management and Analysis
- Vulnerability Detection and Assessment

To learn more about the DHS, Office of Cybersecurity and Communications, go to www.dhs.gov/cybercareers. To apply for a vacant position please go to www.usajobs.gov or visit us at www.DHS.gov.